Package 'InterVA5'

January 20, 2025

Type Package

Title Replicate and Analyse 'InterVA5'

Version 1.1.3

Date 2021-07-19

Depends R (>= 3.5.0)

Encoding UTF-8

Imports curl, readxl

Description Provides an R version of the 'InterVA5' software (<http://www.byass.uk/interva/>) for coding cause of death from verbal autopsies. It also provides simple graphical representation of individual and population level statistics.

License GPL-3

NeedsCompilation no

Repository CRAN

RoxygenNote 7.1.1

Suggests testthat, ggplot2

Author Jason Thomas [aut, cre],
Zehang Li [aut],
Peter Byass [aut],
Tyler McCormick [aut],
Matthew Boyas [aut],
Sam Clark [aut]

Maintainer Jason Thomas <jarathomas@gmail.com>

Date/Publication 2021-07-21 14:30:05 UTC

Contents

InterVA5-package   *Perform InterVA5 algorithm and provide graphical summarization of*
                         COD distribution.

Description

Computes individual cause of death and population cause-specific mortality fractions using the
InterVA5 algorithm. Provides a simple graphical representation of the result.

Details

To get the most up-to-date version of the package, as well as the past versions, please check the
github repository at: https://github.com/verbal-autopsy-software/InterVA5

Package:	InterVA5
Type:	Package
Version:	1.0
Date:	2018-02-01
License:	GPL-3

Author(s)

Jason Thomas, Zehang Li, Tyler McCormick, Sam Clark

Maintainer: Jason Thomas <jarathomas@gmail.com>

References

http://www.byass.uk/interva/

---

causetextV5 *Translation list of COD codes*

Description

This is the translation of COD abbreviation codes into their corresponding full names.

Format

A data frame with the translation of codes to their names for 3 pregnancy statuses, 61 CODs (both the version of COD only and COD with group code), and 6 circumstances of mortality (COMCAT).

Examples

```
data(causetextV5)
```

COMCAT.interVA5 *Summarize population level mortality fraction by Circumstance of Mortality Category*

Description

The function takes input of a list of va object and calculates the mortality fraction by Circumstance of Mortality Category.

Usage

```
COMCAT.interVA5(va)
```

Arguments

va   The list of va object to summarize.

Value

dist.cod   The cause-specific mortality fraction (including undetermined category).

Author(s)

Jason Thomas, Zehang LI, Tyler McCormick, Sam Clark

See Also

CSMF5

Examples

```
## Not run:
data(RandomVA5)
# only fit first 20 observations for a quick illustration
RandomVA5 <- RandomVA5[1:20, ]

sample.output <- InterVA5(RandomVA5, HIV = "h", Malaria = "v",
      write=TRUE, directory = tempdir(),
      filename = "VA5_result", output = "extended", append = FALSE)
## Get CSMF without plots
comcat <- COMCAT.interVA5(sample.output$VA5)

## End(Not run)
```

CSMF.interVA5   *Summarize population level cause-specific mortality fraction as In-*
                             terVA5 suggested.

Description

The function takes input of a list of va object and calculates the cause-specific mortality fraction. It
only calculates CSMF5 as aggregation of up to the third largest causes.

Usage

```
CSMF.interVA5(va)
```

Arguments

va   The list of va object to summarize.

Value

dist.cod   The cause-specific mortality fraction (including undetermined category).

Author(s)

Jason Thomas, Zehang LI, Tyler McCormick, Sam Clark

See Also

CSMF5

Examples

```
## Not run:
data(RandomVA5)
# only fit first 20 observations for a quick illustration
RandomVA5 <- RandomVA5[1:20, ]

sample.output <- InterVA5(RandomVA5, HIV = "h", Malaria = "v", write=TRUE,
        directory = tempdir(), filename = "VA5_result", output = "extended", append = FALSE)
## Get CSMF without plots
csmf <- CSMF.interVA5(sample.output$VA5)

## End(Not run)
```

CSMF5 *Summarize and plot a population level distribution of va probabilities.*

Description

The function takes input of a list of va object and produces a summary plot for the population distribution.

Usage

```
CSMF5(
  va,
  top.aggregate = NULL,
  InterVA.rule = FALSE,
  noplot = FALSE,
  title = "Top CSMF Distribution",
  type = "bar",
  top.plot = 10,
  return.barplot = FALSE,
  min.prob = 0,
  ...
)
```

Arguments

va	The list of va object to summarize.
top.aggregate	Integer indicating how many causes from the top need to go into summary. The rest of the probabilities goes into an extra category "Undetermined". When set to NULL, default is all causes to be considered. This is only used when InterVA.rule set to "FALSE".
InterVA.rule	If it is set to "TRUE", only the top 3 causes reported by InterVA5 is calculated into CSMF as in InterVA5. The rest of probabilities goes into an extra category "Undetermined". Default set to "FALSE".

noplot	A logical value indicating whether the plot will be shown. If it is set to "TRUE", only the CSMF will be returned.
title	A character string for the title of the CSMF plot.
type	An indicator of the type of chart to plot. "pie" for pie chart; "bar" for bar chart.
top.plot	the maximum number of causes to plot in bar plot
return.barplot	A logical indicating if the (barplot) ggplot() object should be returned (instead of printed). Default value is FALSE.
min.prob	The minimum probability that is to be plotted in bar chart, or to be labeled in pie chart.
...	Arguments to be passed to/from graphic function [barplot](), [pie](), and more graphical paramters (see [par]()). They will affect the main title, size and font of labels, and the radius of the pie chart.

Value

dist.cod	The population probability of CODs.

Author(s)

Jason Thomas, Zehang LI, Tyler McCormick, Sam Clark

See Also

[CSMF.interVA5]()

Examples

```
## Not run:
data(RandomVA5)
# only fit first 20 observations for a quick illustration
RandomVA5 <- RandomVA5[1:20, ]

sample.output <- InterVA5(RandomVA5, HIV = "h", Malaria = "v", write = FALSE,
        directory = tempdir(), filename = "VA5_result", output = "extended",
        append = FALSE)

## Get CSMF by considering only top 3 causes reported by InterVA5.
## This is equivalent to using CSMF.interVA5() command Note that
## it's different from using all top 3 causses, since they may not
## all be reported
CSMF.summary <- CSMF5(sample.output, InterVA.rule = TRUE,
   noplot = TRUE)

## Population level summary using pie chart
CSMF.summary2 <- CSMF5(sample.output, type = "pie",
 min.prob = 0.01, title = "population COD distribution using pie chart",
 clockwise = FALSE, radius = 0.7, cex = 0.7, cex.main = 0.8)

## Population level summary using bar chart
```

```
CSMF.summary3 <- CSMF5(sample.output, type = "bar",
  min.prob = 0.01, title = "population COD distribution using bar chart",
  cex.main = 1)
CSMF.summary4 <- CSMF5(sample.output, type = "bar",
  top.plot = 5, title = "Top 5 population COD distribution",
  cex.main = 1)

## End(Not run)
```

DataCheck5   *Data cleaning for InterVA-5 algorithm*

Description

This function implements the data cleaning steps in the InterVA5 software.

Usage

```
DataCheck5(Input, id, probbaseV5, InSilico_check = FALSE, write)
```

Arguments

Input	original data vector for one observation coded by 0 (absence), 1 (presence), and NA (missing).
id	id for this observation
probbaseV5	matrix of probbaseV5
InSilico_check	logical indicator for if the check uses InSilicoVA rule. InSilicoVA rule sets all symptoms that should not be asked to missing. In contrast, the default InterVA5 rule sets these symptoms to missing only when they take the substantive value.
write	logical indicator of writing to file

Value

Output	new data vector
firstPass	message for the first pass check
secondPass	message for the second pass check

Author(s)

Jason Thomas, Zehang Li, Tyler McCormick, Sam Clark

References

http://www.interva.net/

See Also

[InterVA5.plot](#)

Examples

```
data(RandomVA5)
data(probbaseV5)
probbaseV5 <- as.matrix(probbaseV5)
RandomVA5 <- as.matrix(RandomVA5)
input <- as.character(RandomVA5[1, ])
input[which(toupper(input) == "N")] <- "0"
input[which(toupper(input) == "Y")] <- "1"
input[which(input != "1" & input != "0")] <- NA
input <- as.numeric(input)
output <- DataCheck5(Input=input, id="d1", probbaseV5=probbaseV5, write=TRUE)
```

---

getTopSymptoms   *Get the symptoms with the largest conditional probability (symptom |*
                              cause) for causes assigned by InterVA-5.

Description

The function takes an interVA5 object and the data used to assign the causes, and returns the the
symptoms that contribute to the cause assignment (ranked in order of the conditional probabilities
of observing a symptom, given the death is due to that particular cause).

Usage

```
getTopSymptoms(object, data, IDs = NULL, pretty = TRUE, includeAll = FALSE)
```

Arguments

object	An interVA5 object (i.e., the results returned from the InterVA5() function).
data	The input data that InterVA5 used to assign the causes of death.
IDs	A vector that contains the IDs for each death (note that all of IDs are contained in data$ID and object$ID).
pretty	A logical indicating if you want the results in an easy-to-read format (default is 'TRUE')
includeAll	A logical indicating if you want all of the symptoms included in the output (even those which are absent or have a value of missing/no) (default is 'FALSE' which only includes symptoms that are present).

Value

dist.cod   A list of results for each death (organized by ID). For each death, a list is re-
             turned that includes the death's ID, the cause, and a vector of strings listing a
             symptom, it if contributes to the cause assignment (if includeAll = TRUE), and
             the conditional probability of observing the symptom given that the death is due
             to this cause.

Author(s)

Jason Thomas, Zehang LI, Tyler McCormick, Sam Clark

See Also

InterVA5, getTopSymptoms

Examples

```
## Not run:
data(RandomVA5)
sample.output <- InterVA5(RandomVA5, HIV = "h", Malaria = "v", write=FALSE)
topSymptoms <- getTopSymptoms(object = sample.output,
                             data = RandomVA5,
                             IDs = sample.output$ID[1],
                             pretty = TRUE,
                             includeAll = FALSE)

## End(Not run)
```

InterVA5   *Provide InterVA5 analysis on the data input.*

Description

This function implements the algorithm in the InterVA5 software. It produces individual cause of
death (COD) and population cause-specific mortality fractions. The output is saved in a .csv file
specified by user. The calculation is based on the conditional and prior distribution of 61 CODs.
The function can also save the full probability distibution of each individual to file. All information
about each individual is saved to a va class object.

Usage

```
InterVA5(
  Input,
  HIV,
  Malaria,
  write = TRUE,
  directory = NULL,
```

```
    filename = "VA5_result",
    output = "classic",
    append = FALSE,
    groupcode = FALSE,
    sci = NULL,
    returnCheckedData = FALSE,
    ...
)
```

Arguments

`Input`	A matrix input, or data read from csv files in the same format as required by InterVA5. Sample input is included as data(RandomVA5).
`HIV`	An indicator of the level of prevalence of HIV. The input should be one of the following: "h"(high),"l"(low), or "v"(very low).
`Malaria`	An indicator of the level of prevalence of Malaria. The input should be one of the following: "h"(high),"l"(low), or "v"(very low).
`write`	A logical value indicating whether or not the output (including errors and warnings) will be saved to file. If the value is set to TRUE, the user must also provide a value for the parameter "directory".
`directory`	The directory to store the output from InterVA5. It should either be an existing valid directory, or a new folder to be created. If no path is given and the parameter for "write" is true, then the function stops and and error message is produced.
`filename`	The filename the user wish to save the output. No extension needed. The output is in .csv format by default.
`output`	"classic": The same deliminated output format as InterVA5; or "extended": delimited output followed by full distribution of cause of death proability.
`append`	A logical value indicating whether or not the new output should be appended to the existing file.
`groupcode`	A logical value indicating whether or not the group code will be included in the output causes.
`sci`	A data frame that contains the symptom-cause-information (aka Probbase) that InterVA uses to assign a cause of death.
`returnCheckedData`	
	A logical indicating if the checked data (i.e., the data that have been modified by the consistency checks) should be returned.
`...`	not used

Details

Be careful if the input file does not match InterVA5 input format strictly. The function will run normally as long as the number of symptoms are correct. Any inconsistent symptom names will be printed in console as warning. If there is a wrong match of symptom from warning, please change the input to the correct order.

Value

ID	identifier from batch (input) file
MALPREV	selected malaria prevalence
HIVPREV	selected HIV prevalence
PREGSTAT	most likely pregnancy status
PREGLIK	likelihood of PREGSTAT
PRMAT	likelihood of maternal death
INDET	indeterminate outcome
CAUSE1	most likely cause
LIK1	likelihood of 1st cause
CAUSE2	second likely cause
LIK2	likelihood of 2nd cause
CAUSE3	third likely cause
LIK3	likelihood of 3rd cause
COMCAT	most likely circumstance of mortality
COMNUM	likelihood of COMCAT
wholeprob	full distribution of causes of death

Author(s)

Jason Thomas, Zehang Li, Tyler McCormick, Sam Clark

References

http://www.interva.net/

See Also

[InterVA5.plot](#)

Examples

```
data(RandomVA5)
# only fit first 5 observations for a quick illustration
RandomVA5 <- RandomVA5[1:5, ]

## to get easy-to-read version of causes of death make sure the column
## orders match interVA5 standard input this can be monitored by checking
## the warnings of column names

sample.output1 <- InterVA5(RandomVA5, HIV = "h", Malaria = "l", write = FALSE,
    directory = tempdir(), filename = "VA5_result", output = "extended", append = FALSE)

## Not run:
## to get causes of death with group code for further usage
```

```
sample.output2 <- InterVA5(RandomVA5, HIV = "h", Malaria = "l",
  write = FALSE, directory = "VA test", filename = "VA5_result_wt_code", output = "classic",
    append = FALSE, groupcode = TRUE)

## End(Not run)
```

InterVA5.plot   *Plot an individual-level distribution of va probabilities.*

Description

The function takes an input of a single va object and produces a summary plot for it.

Usage

```
InterVA5.plot(
  va,
  type = "bar",
  title = "Top CSMF Distribution",
  min.prob = 0.01,
  ...
)
```

Arguments

va	A va object
type	An indicator of the type of chart to plot. "pie" for pie chart; "bar" for bar chart.
title	A character string for the title of the CSMF plot.
min.prob	The minimum probability that is to be plotted in bar chart, or to be labeled in pie chart.
...	Arguments to be passed to/from graphic function barplot, pie, and more graphical paramters (see par). They will affect the main title, size and font of labels, and the radius of the pie chart.

See Also

CSMF5

Examples

```
## Not run:
data(RandomVA5)
# only fit first 20 observations for a quick illustration
RandomVA5 <- RandomVA5[1:20, ]
#' sample.output <- InterVA5(RandomVA5, HIV = "h", Malaria = "v", write = FALSE,
```

```
        directory = tempdir(), filename = "VA5_result", output = "extended", append = FALSE)

## Individual level summary using pie chart
InterVA5.plot(sample.output$VA5[[3]], type = "pie", min.prob = 0.01,
    main = "1st sample VA analysis using pie chart", clockwise = FALSE,
    radius = 0.6, cex = 0.6, cex.main = 0.8)


## Individual level summary using bar chart
InterVA5.plot(sample.output$VA5[[3]], type = "bar", min.prob = 0.01,
    main = "2nd sample VA analysis using bar chart", cex.main = 0.8)

## End(Not run)
```

---

print.interVA5_summary

Print method for summary of the results obtained from InterVA5 algorithm

---

Description

This function prints the summary message of the fitted results.

Usage

```
## S3 method for class 'interVA5_summary'
print(x, ...)
```

Arguments

x   summary of InterVA5 results

...   not used

---

probbaseV5   *Conditional probability of InterVA5 (version 17 – Sept. 9th, 2018)*

Description

This is the table of conditional probabilities of symptoms given CODs, along with prior probabilities in the first row. The values are from InterVA-5

Format

A data frame with 354 observations on 87 variables. The first row contains observations corresponding to prior probabilities; while the subsequent observations (rows 2 - 354) are the conditional probabilities.

Examples

```
data(probbaseV5)
```

---

probbaseV5_14	*Version 14 of the conditional probability of InterVA5*

Description

This is version 14 (February 15th, 2018) of the table of conditional probabilities of symptoms given CODs, along with prior probabilities in the first row. The values are from InterVA-5

Format

A data frame with 354 observations on 87 variables. The first row contains observations corresponding to prior probabilities; while the subsequent observations (rows 2 - 354) are the conditional probabilities.

Examples

```
data(probbaseV5_14)
```

---

probbaseV5_17	*Version 17 of the conditional probability of InterVA5*

Description

This is version 17 (Sept. 9th, 2018) of the table of conditional probabilities of symptoms given CODs, along with prior probabilities in the first row. The values are from InterVA-5

Format

A data frame with 354 observations on 87 variables. The first row contains observations corresponding to prior probabilities; while the subsequent observations (rows 2 - 354) are the conditional probabilities.

Examples

```
data(probbaseV5_17)
```

| probbaseV5_18 | *Version 18 of the conditional probability of InterVA5* |

Description

This is version 18 (April 3, 2020) of the table of conditional probabilities of symptoms given CODs, along with prior probabilities in the first row. The values are from InterVA-5

Format

A data frame with 354 observations on 87 variables. The first row contains observations corresponding to prior probabilities; while the subsequent observations (rows 2 - 354) are the conditional probabilities.

Examples

```
data(probbaseV5_18)
```

| probbaseV5_19 | *Version 19 of the conditional probability of InterVA5* |

Description

This is version 19 (July 20, 2021) of the table of conditional probabilities of symptoms given CODs, along with prior probabilities in the first row. The values differ from the last version (v18) of InterVA-5 (interva.net) by setting Pr(abortion-related death | i309 = 1) = "N" Pr(abortion-related death | i310 = 1) = "N" (the previous values were "E").

Format

A data frame with 354 observations on 87 variables. The first row contains observations corresponding to prior probabilities; while the subsequent observations (rows 2 - 354) are the conditional probabilities.

Examples

```
data(probbaseV5_19)
```

RandomVA5	*200 records of Sample Input*

Description

This is a dataset consisting of 200 arbitrary sample input deaths in the acceptable format of InterVA5. Any dataset that needs to be analyzed by this package should be in the same format. The order of the input fields must not be changed.

Format

200 arbitrary input records.

Examples

```
data(RandomVA5)
```

summary.interVA5	*Summary of the results obtained from InterVA5 algorithm*

Description

This function prints the summary message of the fitted results.

Usage

```
## S3 method for class 'interVA5'
summary(object, top = 5, id = NULL, InterVA.rule = TRUE, ...)
```

Arguments

object	fitted object from InterVA5()
top	number of top CSMF to show
id	the ID of a specific death to show
InterVA.rule	If it is set to "TRUE", only the top 3 causes reported by InterVA5 is calculated into CSMF as in InterVA5. The rest of probabilities goes into an extra category "Undetermined". Default set to "TRUE".
...	not used

References

http://www.interva.net/

Examples

```
## Not run:
data(RandomVA5)
# only fit first 20 observations for a quick illustration
RandomVA5 <- RandomVA5[1:20, ]

## to get easy-to-read version of causes of death make sure the column
## orders match interVA5 standard input this can be monitored by checking
## the warnings of column names

sample.output1 <- InterVA5(RandomVA5, HIV = "h", Malaria = "l",
    write = FALSE, directory = tempdir(), filename = "VA5_result",
    output = "extended", append = FALSE)

summary(sample.output1)
summary(sample.output1, top = 10)
summary(sample.output1, id = "sample3")

## End(Not run)
```

---

whyNotCOD	*Get the symptoms with the largest conditional probability (symptom | cause) using VA data.*

Description

The function takes takes verbal autopsy data (which can be passed to InterVA5() to assign causes of death), and returns the the symptoms that contribute to the assignment of a particular cause of death. This function differs from getTopSymptom() in that the user specified the cause for which they would like the results. This is an interactive function in the sense that if a cause is not provided as an argument, then the function will print out a numbered list of possible causes and the user can enter in the number to identify the cause of interest.

Usage

```
whyNotCOD(data, IDs = NULL, cause = NULL, pretty = TRUE, includeAll = FALSE)
```

Arguments

data	The input data that InterVA5 used to assign the causes of death.
IDs	A vector that contains the IDs for each death (note that all of IDs are contained in data$ID and object$ID).
cause	A string giving the name of the cause for which the conditional probabilities will be returned.
pretty	A logical indicating if you want the results in an easy-to-read format (default is 'TRUE').
includeAll	A logical indicating if you want all of the symptoms included in the output (even those which are absent or have a value of missing/no) (default is 'FALSE' which only includes symptoms that are present).

Value

dist.cod   A list of results for each death (organized by ID). For each death, a list is re-
                turned that includes the death's ID, the cause, and a vector of strings listing a
                symptom, it if contributes to the cause assignment (if includeAll = TRUE), and
                the conditional probability of observing the symptom given that the death is due
                to this cause.

Author(s)

Jason Thomas, Zehang LI, Tyler McCormick, Sam Clark

See Also

[InterVA5](), [whyNotCOD]()

Examples

```
## Not run:
data(RandomVA5)
whyNotCOD(data = RandomVA5,
          IDs = RandomVA5$ID[1],
          pretty = TRUE,
          includeAll = FALSE)

data(causetextV5)
causetextV5[22, 2]
whyNotCOD(data = RandomVA5,
          IDs = RandomVA5$ID[1],
          cause = causetextV5[22, 2],
          pretty = TRUE,
          includeAll = FALSE)

## End(Not run)
```

Index